**For machine Learning:**

**K-Fold score or cross validation:**

from sklearn.model\_selection import cross\_val\_score

obj = cross\_val\_score(model\_name,featurs,label,cv=num)

**Features Selection (Features importances technique):**

From sklearn.ensemble import ExtraTreesClassifier

Model\_obj = ExtraTreesClassifier()

Model\_boj.fit(features,level)

Model\_obj.feature\_importances\_

**Encoding technique:**

1. dataset["columns\_name"] = LabelEncoder().fit\_transfrom(dataset["column\_name"])
2. encoder = LabelEncoder()

x = dataset.apply(encoder.fit\_transform)

1. for column in features.columns:

if features[column].dtype == np.number:

continue

features[column] = encoder.fit\_transform(features[column])

**Feature Scaling technique:**

1. sc = StandardScaler()

sc.fit\_transform(xtrain)

sc.transform(xtest)

**when load data from sklearn than if we can see those data as a data frame than use this technique:**

1. data = datasets.load\_boston()

df = pd.DataFrame(data.data,columns=data.feature\_names)

df[“target”] = data.target

**For Pandas:**

dataset = dataset.fillna(dataset.mean()) 🡪 filling missing value

to see which rows contain missing value in data frame

numerical\_df[numerical\_df[num\_var\_miss].isnull().any(axis=1)]

to see specifically rows contain missing value in data frame

numerical\_df[num\_var\_miss] [numerical\_df[num\_var\_miss].isnull().any(axis=1)]

Finding the numerical columns:

1. numerical\_col = x\_train.select\_dtypes(include=["int64","float64"]).columns